

AudioSession API (Youenn, 20 minutes)

High-level presentation

- Reviving the Audio Focus API
 - Explainer available at <https://github.com/youennf/audio-focus/blob/main/explainer.md>
- Focus on low hanging fruits for Android/iOS platforms
 - AudioSession category and interruption detection
- Create a foundation for future features
 - Request/abandon audio focus explicitly
 - Group/control media elements as a whole
- Initial prototype (off by default) in WebKit

Example: setting AudioSession category

- <https://jsfiddle.net/wp4o2yqL/> on iOS
 - Calling capture then playAudio is fine
 - Calling playAudio then capture is disruptive
 - Calling setCategory beforehand fixes the disruption

```
function playAudio() {  
  const context = new AudioContext();  
  const oscillator = context.createOscillator();  
  oscillator.type = 'square';  
  oscillator.frequency.setValueAtTime(440, context.currentTime);  
  oscillator.connect(context.destination);  
  oscillator.start();  
}
```

← May trigger 'ambient'

```
function capture() {  
  navigator.mediaDevices.getUserMedia({ audio: true });  
}
```

← May trigger 'play-and-record'

```
function setCategory() {  
  navigator.audioSession.type = "play-and-record";  
}
```

Example: handling of audio interruption

- Custom resume of capture & playing after interruption

```
navigator.audioSession.type = 'play-and-record';
remoteVideo.srcObject = grabRemoteMediaStream();
localVideo.srcObject = await navigator.mediaDevices.getUserMedia(
  { audio:true, video:true });
```

Start the application, play media and capture

```
let isInterrupted = false;
navigator.audioSession.onstatechange = async () => {
  if (navigator.audioSession.state === 'interrupted') {
    isInterrupted = true;
    localVideo.pause(); remoteVideo.pause();
    localVideo.srcObject.getTracks().forEach(track => track.enabled = false);
    showInterruptedBanner();
  } else if (isInterrupted) {
    isInterrupted = false;
    await showRestartBanner();
    localVideo.srcObject.getTracks().forEach(track => track.enabled = true);
    localVideo.play(); remoteVideo.play();
  });
}
```

When interrupted, pause media playing and capture + notify user

When uninterrupted, notify user and restart media playing and capture based on user input

API overview

- WebIDL

```
enum AudioSessionState { "inactive", "active", "interrupted" };
```

```
enum AudioSessionType { "auto", "playback", "transient",  
    "transient-solo", "Ambient", "play-and-record" };
```

```
[Exposed=Window]
```

```
interface AudioSession : EventTarget {  
    attribute AudioSessionType type ;  
    readonly attribute AudioSessionState state ;  
    attribute EventHandler onstatechange ;  
};
```

```
[Exposed=Window]
```

```
partial interface Navigator {  
    // The default audio session that the user agent will use  
    readonly attribute AudioSession audioSession ;  
};
```

This is just the start

- Some issues already filed
 - More are welcome
- Feedback most welcome, in particular
 - Is it the right MVP?
 - What are the next steps?